Cider PRESS

San Francisco Apple Core

	Page
REM	3
Coming in May -Michael Fischer	3
April Meeting Notes -Bill Neff	4
Conflicting Technologies	4
Pascal - FileFinder -Max Nareff	5
Notice: Market Research Testing	5
Positions Available	5
Contest Entry Form	6
Programming Contest Update -Don Scellato	7
Marketing Dept. Notes -Don Scellato	7
Pascal Hint -Michael Fischer	8
BASIC -Joyce Conklin	9
Reviews:	
M.I.T.E. -Woody Liswood	11
Punctuation + Style -Woody Liswood	14
Error Traps	16
Pascal -Mike Schnapp	16
Reviews:	
Soft-Step -Joyce Conklin	17
Karel the Robot -Michael Fischer	18
Interest Groups	20
Interest Group Calendar	21
Apple Core Offerings	22

GENERAL MEETINGS

MAY 21
JUNE 18
#JULY 9

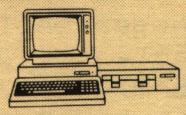
MAY, 1983

His Master's New Voice.

The Computer Age is here. We're all caught in the middle of a communications revolution. Business problems are more complex and so is the equipment you need to solve the problems. Cybet has the answers. Top quality lines of personal computers including hardware, software and warranty-backed service. A knowledgeable staff whose advice you've come to trust since 1980 when Cybet started as Systems Technology. Let Cybet help you make the transition to the Computer Age. Cybet, the store with the answers.

KAYPRO II

apple


NEC

Cybet Computers
Burton Village
2270-I Oak Grove Road
Walnut Creek, CA 94598
(415) 933-1131

THE CORE'S CORE

DIRECTORS:

Sami Capron	415/388-8631
Joyce Conklin	415/341-6955
Michael Fischer	415/933-4435
Gary Knox	415/521-0603
Stepehn C. Lloyd	415/222-5860
Don Scellato	415/672-2051
Alan L. Smerican	415/566-2342

MEMBER SERVICE/COMMITTEES:

Membership		
Alan L. Smerican		415/566-2342
	Ted Chatterton	
	Sami Capron	
Cider Press -		
Acting Editor	Allan Woods	415/687-9659
Graphics	Jim Linhart	415/652-0909
BASIC	Joyce Conklin	415/341-6955
Int. Grp. Coord.	Sami Capron	415/388-8631
Marketing Dir.	Don Scellato	415/672-2051
Review Group	("Woody") Liswood	415/585-6375
Demo Coord'r	Rance Fields	415/563-2173
General Meeting	Michael Fischer	415/933-4435
Exec. Secretary	Stephen C. Lloyd	415/222-5868
Treasurer	Gary Knox	415/521-0603
Coordinating Dir.	Joyce Conklin	415/341-6955

CIDER PRESS POLICY

All manuscripts, photographs, and other materials will be considered submitted free and released for publication. They become the property of the San Francisco Apple Core and the CIDER PRESS. Authors should clearly mark all material submitted for publication so that proper credit can be given.

The views expressed in any articles are those of the authors, and not necessarily those of the San Francisco Apple Core, Inc., the Editor, Club Officers or Directors . . . or anyone else in their right minds.

The San Francisco Apple Core Incorporated is a non - profit organization comprised of and supported by owners and users of Apple I I and Apple / / / computers. The Club is run by officers and committees, and the Club endeavors to aid Apple owners through its Program Libraries, Program Disks, meetings, and this publication.

Article Submission: The CIDER PRESS welcomes any and all articles dealing with Apple computers and their associated hardware and software. Articles may be submitted (typed or printed, double - space), or as disk text files using any of the popular word processors or text editors. Any author who has an article published in the CIDER PRESS will receive a free copy of the most current Disk - of - the - Month.

Reprints: All articles appearing in the CIDER PRESS not copyrighted by the author may be reprinted by other *non - profit* Apple User Groups, without prior permission, so long as proper credit is given to both the author and the CIDER PRESS. Permission to reprint a copyrighted article must be obtained in advance, by writing to the authors, c/o the San Francisco Apple Core.

San Francisco Apple Core
1515 Sloat Blvd.
San Francisco, CA 94132

CIDER PRESS

REM The CIDER PRESS has a new editor. We (editors call themselves "we") are growing into the job. Changes in the C.P. will be gradual to avoid throwing the baby out with the bathwater.

The CIDER PRESS is a reflection of the club's vitality and activity. It is the result of the combined effort of those who wish to contribute to it and be a part of it. (No one is paid. We share the work and the pizza.) We would hope that you think of it not as a magazine from a paid subscription, but rather as a BBS (Bulletin Board Service) which you don't need a MODEM to access.

The APPLE CORE is people grouped to share information. The CIDER PRESS is one vehicle for that sharing and is open to you. We welcome any and all of you to join in, participate and enjoy. Ask questions, submit material, make suggestions - after all, without you there would be no CIDER PRESS.

Spring is a time of rebirth and rejuvenation and I hope to see a flowering of Apple blossoms in the coming season.

-Editor.

■■■■■■■■■■■■■■■

By Michael Fischer

Coming in May

With the general meeting set for May 21 we will be inagurating our *something for the entire family* format. So bring your entire family, especially your children.

DesignWare, an educational software design and marketing company, will present demonstrations of some of their educational products.

Broderbund Software will show some of their games such as *Arcade Machine* and *Choplifter* as well as their innovative word processor *Bank Street Writer*.

DesignWare

DesignWare has produced sfotware for SRA, Reader's Digest, Harper & Row, and Spinnaker among others. They developed Facemaker and Story Machine (ranked 3rd and 5th in Softalk's best selling home education program list for April). They also write programs for older students such as Problem Solving Strategies.

The company has recently released some software under their own name, including two spelling word games they will demonstrate at the meeting, Cryptocube and Spellicopter.

Apple Core members with kids in 2nd to 8th grade can take advantage of Designware's market research sessions to obtain free software. This opportunity will be discussed at their presentation.

Broderbund

Broderbund needs no introduction for Apple arcade game players. Choplifter has been on Softalk's top 10 arcade program for a number of months; Arcade Machine ranks 6th and Seafox 9th on the same list for April.

Less well known is the company's other software. They have published business and accounting packages. Broderbund also publishes one of the hotest word processors, Bank Street Writer. This program was touted by Time Magazine as the word processor for those afraid of word processing.

Kid's Corner

For May we plan to have a demonstration geared for kids as well as one designed for adults.

Beginning in June a separate kid's corner will be inaugurated (upstairs with the sales table). We hope to have kids bring in programs they like and demonstrate them to other kids. Other acitivities as planned by the kids would be available.

[I would appreciate one or more adults who would volunteer to assist at this special area. If enough people come forward the task will be light for everyone.]

Future Meetings

Future plans include not only demonstrations of the practical and impractical (including robots, mice, graphics programs, databases, and much more) by both the publishers and club member users, but also talks on various subjects. I welcome your suggestions on subjects, speakers, and products, as well as your comments on the meetings.

April Meeting Wrapup

Thanks to our three demonstrators at the April meeting:

1. Ed Badasov of FOLLK (Friends of LISP, Logo & Kids) who introduced us to the logo language and such subjects as turtle graphics and recursion.

2. Ed Swartzman of Voice Recognition Systems who showed us a voice input module which accepts voice commands to your programs; and

3. Mike Nadelman of Apex Systems who showed us the portable Atrona Attache computer which can exchange data with an Apple.

See you May 21 at Room 100, Building C, Fort Mason. We will start at 10 a.m. sharp.

SAN FRANCISCO APPLE CORE
GENERAL MEETING
April 16, 1983

President Mike Fischer called the meeting to order at 10:07:22 (according to Woody's watch) before a crowd of about 80 at Ft. Mason. He announced the new (actually the old, old) meeting format which will be more member oriented and then introduced the other board members, group leaders, etc. See elsewhere in this issue for names and numbers. Also, consult the interest group schedule for time and place of the various interest group meetings.
Allan described the new format of the Cider Press, which also we be more of a club in-house organ than a computer journal (which is what we used to have). (Sec note: If you don't have one, get a copy of "Best of the Cider Press 1978-1979". It's full of notes, short routines, questions, and all kinds of usefull stuff for anyone writing in BASIC. "Best of . . . 1980" has lots of stuff for PASCAL users.)
Don made several announcements concerning sales, discs, program contest, visicalc meetings and so forth. John Uhley (one of the few old-timers at the meeting) announced his course on program security to be offered at the Marin Computer Center starting May 15. Call John at 461-7626 for more info. Lorraine of S.E. education needs a source for renting computers for her upcoming courses. Call 468-0456 if vou have any leads. Contact Mike if you are interested in a young peoples' interest group. (Mike, all of us computer buffs are young; it's just that those with fewer wrinkles relate to the machines easier.)
ALSO! We are returning to the format of having members give demos at the meeting, so line up! (We always have had this one, maybe we can have "show and tell" meetings with everybody who wants to assigned to a corner to do their thing. Several of our members got their commercial start that way.) I'm kind of using these meeting notes to capitalize on Mike's suggestion for meeting, group, Press, etc. suggestions. Join in!
Demonstrations:
A representative from FOLK gave us a beautiful demonstration of the APPLE LOGO language which is a form of list processing language used by the A(rtificial) I(ntellegence) researchers for problem solving. (See the last 2 issues of Scientific American.) LOGO is super for the young people that FOLK works with because 1) it fits their logic processing and 2) it is highly rewarding. We had a ball watching the turtle graphics portion develop a problem solving graphics situation, THEN learned how LOGO allows you (logically) to reduce your multiline program to ONE SINGLE LINE. The next part of our demo was the use of the powerful list processing capabilities. Then an interesting graphics problem and solution which is titled "anygon" (that's geometry?).
You have to hear this one to believe it!
AVIM means Apple Voice Input Module.
You "train" it to recognise your voice for the commands to operate a program or collect data or compute your inventory or where ever your needs and imagination lead. Just close the switch (hand or foot), say the word and you've got it.
Grab it quick before it goes national and the price jumps. 12:11:07.

Bill Neff

rec sec

CONFLICTING TECHNOLOGIES

If you are thinking about getting a MODEM, or if you have one and are thinking of enhancing your telephone service with the addition of Call-Waiting, you should be aware that the momemtary interruption and the notifying tone that tells you someone is trying to call you have a disruptive effect on MODEM communication.

The disruption could cause loss of data, incomplete transmission or even disconnect you. Call-Waiting is a destructive disadvantage to MODEM communication. Consider the choices before you act.

A Pascal Procedure - FileFinder

The following procedure can be useful when an input file is required. It ensures that the file will be found and opened if the filename is correct, regardless of it's location on-line. A global,"FileName:String;" must be declared and the procedure is called after the prompt,"Enter Filename ",is answered.

```
READLN (FileName);
FileFinder (FileName);
Etc.

PROCEDURE FileFinder (VAR S:STRING);
VAR
  TempName, Vol : STRING;
  Result    : INTEGER;

  PROCEDURE FileSeeker;
  BEGIN
     (*$I-*)
     TempName := CONCAT (Vol, S);
     RESET (F, TempName);
     Result := IORESULT
     (*$I+*)
  END; (* FileSeeker *)

(* Erases volume names for searching *)
  PROCEDURE EraseVolNames;
  BEGIN
    IF POS ('*' ,S) <> 0 THEN
      S := COPY (S, POS ('*',S) + 1,
            LENGTH (S)-1)
    ELSE
      IF POS(':',S) <> 0 THEN
        S := COPY (S, POS (':', S) + 1,
          LENGTH (S)- POS (':', S))
  END; (* EraseVolNames *)

BEGIN  (* FileFinder *)
  EraseVolNames;
  IF (POS ('.TEXT',S) = 0) THEN
    S := CONCAT (S,'.TEXT');
  TempName := S;
     (* Try boot disk first *)
  RESET (F,TempName);
  IF (IORESULT <> 0) then
    BEGIN
      Vol := '#12:';
      FileSeeker;
      IF (Result <> 0) THEN
        BEGIN
          Vol := '#11:';
          FileSeeker
        END;
      IF (Result <> 0) THEN
        BEGIN
          Vol := '#5:';
          FileSeeker
        END;
      IF (Result <> 0) THEN
        BEGIN
          GOTOXY (10, 12);
          WRITELN
('File not on-line;Is Filename correct?');
          EXIT (PROGRAM)
        END
    END
END; (* FileFinder *)
```

NOTICE

COMPUTER FAMILIES

DesignWare, Inc., an educational software company in San Francisco, is looking for computer families with kids in grades 2 through 8 to participate in market research sessions.

Their on-going testing sessions are held to test new products, advertising and packaging ideas. Registered families will be contacted and afforded the opportunity to participate. Parents are not required to attend with their kids, however there will be some special sessions with parents.

FREE SOFTWARE will be given to all who participate in these sessions.

To sign up, call (415) 546-1866 and ask for Virginia regarding market research testing.

DesignWare will be giving a demostration of two of their newest products, Spellicopter and Crypto-Cube at the may meeting. Company representatives will be on hand to answer questions about their programming, teaching strategies and educational objectives. If you have seen any of their previous works (FaceMaker, StoryMachine, etc.) you'll want to bring your whole family to the club meeting May 21.

DesignWare has created 5-10% of all the software in the institutional software market. They produce software for a wide range of computers, so tell your friends and neighbors about the market testing and the May demonstration. Don't miss it!

++++++++++++++++

POSITIONS AVAILABLE

The CIDER PRESS has openings for stringers, staff writers, investigative reporters, copy persons, graphic artists, advertising reps., proof readers, and verbalists. Apply in person, by mail or by phone to the editor. The CIDER PRESS is an equal opportunity opportunity.

SAN FRANCISCO APPLE CORE

PROGRAMING CONTEST

CONTEST ENTRY FORM

SUBMIT TO: San Francisco Apple Core
1515 Sloat Blvd., Suite 2
San Francisco, CA 94132
Attn: Contest Librarian

Entrant name: _____

Address: _____

City: _____ State: _____ Zip: _____

Telephone Number: _____

Program Name: _____

Program Source (If original, so state. If not original, identify source and attach copy of listing of both programs) _____

_____

_____

Description of Program: (What does it do? How?) _____

_____

_____

_____

_____

_____

_____

_____

_____

(Use additional support sheets if required.)

System Requirements: (Describe memory, hardware, language, DOS, or other similar requirements.) _____

_____

_____

_____

DISCLAIMER:

I HEREBY DECLARE THAT THE INFORMATION PROVIDED ABOVE IS TRUE. I HEREBY HOLD HARMLESS THE "SAN FRANCISCO APPLE CORE" FOR ANY ACTION ARISING FORM THE "APPLE CORE'S" DISTRIBUTION OR USE OF THIS PROGRAM. THIS PROGRAM IS DONATED FOR SOLELY EDUCATIONAL PURPOSES. NO WARRANTY FOR ANY OTHER USE IS ASSUMED OR INTENDED.

Entrant Signature: _____

Date: _____ Membership Expiration Date (See Card): _____

SPACE BELOW FOR CLUB USE ONLY:

Date Received: _____ Contest Category Assigned _____

Date Receipt Acknowledged: _____

Reviewers Assigned: _____

_____

_____

_____

Points Assigned by Reviewer 1: _____ Reviewer 2: _____ Reviewer 3: _____

Reviewer 4: _____ Total Points Received: _____

PROGRAMMING CONTEST UPDATE

by

Don Scellato

A copy of the entry blank for the Programming Contest has been included in this issue of The Cider Press for the convenience of Club members. A Press Release has gone out to a number of national circulation computer magazines. The rules for the Contest and a list of program categories was published in the March issue of the Cider Press. You can obtain copies of that article by calling the clubhouse or by mailing a request. Susan will mail copies of that article along with entry blanks to anyone who asks.

The Contest Librarian and board members are still hard at work seeking hardware and software prizes. Although many of the prizes listed below were included in last month's update, I will continue to list them this month along with the new additions. The prizes received or promised to date are listed below along with the name of the firm which has agreed to become a Contest Sponsor by donating a prize or prizes.

1. APPLE COMPUTER, INC., Cupertino, Ca. - Apple Computer, Inc. has supplied a copy of Apple LOGO, a copy of CHILDREN'S TELEVISION WORKSHOP (The Muppet Program), and a copy of NEWS AND QUOTES to be used as prizes in our contest. The first two products have educational applications and the third product will be useful to investors. The SFAC thanks Apple Computer, Inc. for their support.

2. ARTSCI, Inc., 5547 Satsuma Avenue, North Hollywood, Ca., publishers of Magic Window, has contributed two prizes to the SFAC for the programming contest. The products contributed are Magic Window II, a 40 column/70 column/80 column Word Processor, and Magicalc, a very powerful and flexible electronic spreadsheet. We will list the category in which these prizes will be awarded when all of the prizes have been received. The San Francisco Apple Core thanks ARTSCI, Inc. for their generous support in our effort to encourage member programming.

3. The International Apple Core, Apple Orchard, 910A George Street, Santa Clara, Ca., has declared that it will give to each contest entrant a free one year subscription to the Apple Orchard, The Premier Magazine for Apple Computer Users.

Any entrant who already subscribes to this magazine will receive an extension to his subscription. The SFAC thanks the Editor and Publisher for his support and encouragement with the contest.

4. LEGEND INDUSTRIES - This firm has generously agreed to provide a 128K RAM expansion card for use as a prize in the Programming Contest. The SFAC thanks Legend Industries for their support.

5. MICRO-SPARC, INC. - Micro-sparc, Inc. has provided a copy of LEXICOM a utility program which converts word processor files from one word processor to another. The SFAC thanks Micro-Sparc, Inc. for their support of the contest.

6. Software Publishing Corporation, 1901 Landings Mountain Drive, Mountain View, Ca., has contributed a copy of PFS GRAPH to the club for use as a prize in the programming contest. This program is a very useful graphing tool which allows the user to access files from PFS:FILE or Visicalc. It also accepts data entry from the user and provides him with a variety of graphs. The SFAC thanks Software Publishing Corporation for their continuing support.

7. Roger Wagner of Southwestern Data Systems, P.O. Box 582, Santee, Ca., has kindly contributed a copy of The Routine Machine to the Club for use as a contest prize. This programming utility gives the user the ability to access the power and speed of machine language routines in Applesoft programs. High Resolution graphics and sound are placed at the users fingertips in a convenient fashion through this program.

A number of other firms have been approached regarding donating prizes for the contest. All of these firms have indicated a willingness to consider donating prizes but have not yet committed. When they do we will ad their names to the list of Contest Sponsors listed above. In the meantime, the SFAC is most fortunate to receive the support already provided by the companies listed above. I'm sure the membership of the Club joins the Program Librarian and the Board of Directors in extending or thanks.

With the list of prizes growing longer, now is the time to finish up that special program you've been working on and send it in. We'd be glad to get a copy of your latest mortgage amortization schedule, your latest gardening planner, weather forecaster, decision maker, or whatever. So finish it or dust off a program you've been using for a while, dress it up, and send it in.

MARKETING DEPARTMENT NOTES

by

Don Scellato

As soon as I have completed a definitive catalog of the programs contained on the Club's collection of IAC disks, I will place them in a Library for the use of Club members. This Library of IAC disks will be kept at the Clubhouse for copy by those members who wish to make the trip.
The Epson Printer group is working o a Printer Utility Disk which will be ready for sale later this year. The Visicalc Interest Group is working on a second

collection of VisiCalc templates which will be issued later in the year.
We are shopping for a new supply of blank disks and should have some available by the May Meeting.
If there are any products the members would like to see for sale at the table, or if any member would like me to try to obtain any commercial hardware or software products for use with the Apple, give me a call, drop me a note, or see me at the next meeting.
The Marketing Department is seeking individuals to help man marketing tables and to sign up members at the various small and large swap meets which seem to be popping up everywhere in the Bay Area these days. A Sign Up sheet will be provided at the next meeting for anyone who is interested. The Department is also seeking volunteers to help with overall marketing matters and with the process of compiling programs into special application disks and monthly disks. Anyone who is interested should call me or should see me at the next meeting.

SOFTWARE HEADQUARTERS

Whatever your microcomputer needs, The **Softwaire Centre** has it. Featuring the largest selection of **CP/M** business and utility programs available anywhere.

Your Headquarters for:

• CP/M • APPLE • TRS 80 • ATARI • IBM • TECHNICAL BOOKS/MAGAZINES • GAMES N' FUN • COMPUTER SUPPLIES / PAPER / BLANK DISKS • SEMINARS • *BUSINESS PROGRAMS OUR SPECIALTY!*

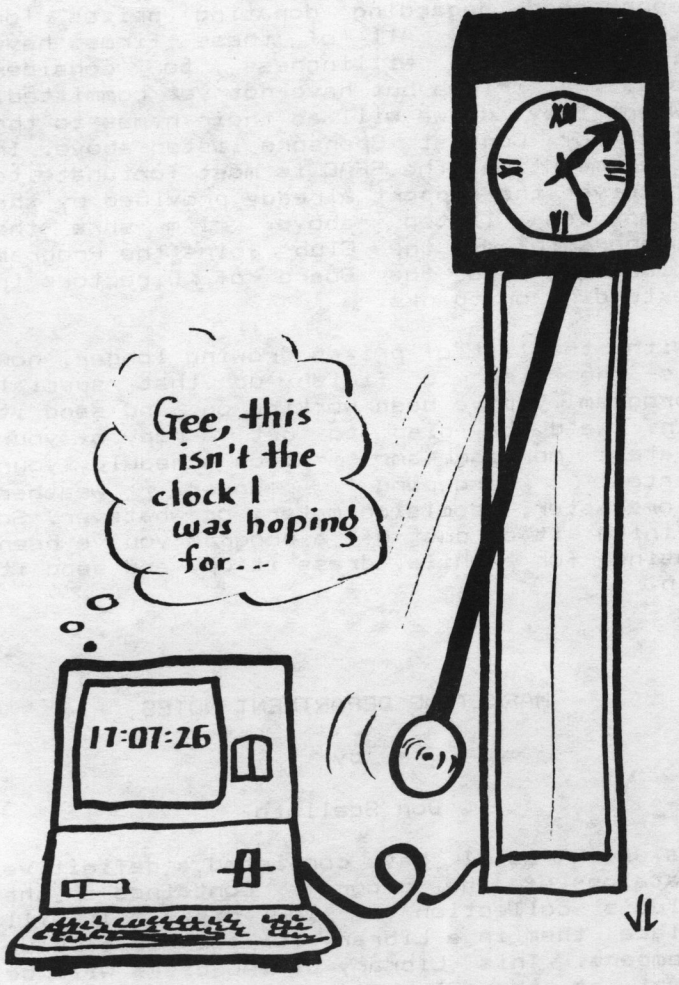
Call for FREE Newsletter.

5% Discount To Anyone Displaying A Current Apple Core Membership Card.

5269 Broadway at College
Oakland • (415) 928-9333
M-F 10-7 • Sat. 10-6 • Sun. 12-5

The Original Softwaire Store™

Softwaire Centre INTERNATIONAL



PASCAL HINT
by Michael Fischer

In BASIC the GET A$ command reads the keyboard and faithfully saves whatever key was pressed. In Pascal the command READ (KEYBOARD, Variable), does the same thing ... EXCEPT when <RETURN> is pressed.
The ASCII value of A$ when <RETURN> is pressed following GET A$ is 13 (Control M or RETURN). The ASCII value of Variable when <RETURN> is pressed following READ (KEYBOARD, Variable) is 32 (Space or blank).
Fortunatelythere is a way around this problem in Pascal. A function can be designed which returns the correct ASCII value of all keys pressed except Control-A, Control-F, Control-S, and Control-Z. These four keys are intercepted by the Pascal System for its own use.

```
FUNCTION GET ASCII:Integer;
  Var
     Inblock: Packed Array [1..2]
              of Integer
              (*Unit read requires
               a packed array*)
  Begin
     Inblock[1]:=0
     Inblock[2]:=0
     Unit read (2, Inblock, 1);
               (*Read 1 character
                from keyboard (unit
                2) into Inblock*)
     GET ASCII:= Inblock[1]
  End;
```

B*A*S*I*C

(c) Copyright by Joyce Conklin, 1983

Finally, I've found time to write the column on sorting which I promised two months ago. (I know, you've all been hanging by your fingernails in desperate anticipation! You can relax now.) Sorting is done with the elements in an array, so that they can be arranged in order, either low to high or vice versa. You can also sort string variables, which will enable you to alphabetize a list. This will follow the same algorithm, but will obviously take longer, since you have to compare each character of the string to the corresponding character of the other string, to decide which one should come first.
For purposes of demonstration we will have a short array of five numbers...

```
10 FOR L = 1 TO 5
20 READ A(L)
30 NEXT L
40 DATA 9,7,3,1,4
```

When this portion of the program is run you will have an array loaded with the five numbers. In order to understand easily what is going to happen with each of these numbers, it is helpful to view the array as a series of "mailboxes" with labels and contents as shown in Fig. 1:

9	7	3	1	4
A(1)	A(2)	A(3)	A(4)	A(5)

FIG. 1

The first sorting routine to be studied will be the bubble sort, so called because the smallest (or largest, depending upon your desired order) number "bubbles" to the top of the array. While it is the easiest to understand, trace, and explain, it is the slowest of all the sorting algorithms. For a very short list, such as our sample, the timing is not critical, but for longer lists (anything over 10 items) it is dreadfully slow. When I was sorting 55-60 student scores using the bubble sort, I could go out and load the dishwasher after dinner while the sort was working, before the printer would start to clatter; about 3-5 minutes. If you want the sort to move faster, you will have to use other algorithms, which are published in various sources. I will also list the version of the Shell-Metzner sort which I now use in my grading program (cf. The Apple Orchard, Mar. 1983).

The bubble-sort algorithm is as follows:

```
100 F = 0
110 FOR L1 = 1 TO 4
120 IF A(L1) <= A(L1+1) THEN 170
130 T = A(L1)
140 A(L1) = A(L1 + 1)
150 A(L1 + 1) = T
160 F = 1
170 NEXT L
180 IF F = 1 THEN F = 0: GOTO 110
```

F in line 110 stands for flag; if the flag = 1 it means that a swap has been made. In line 120, if your array is longer than 5, of course the 4 must be replaced by a number 1 less than the length of your array. If only one swap was made during the first pass through the array, F will be set at 1, and line 180 will allow you to reset F = 0 and try again. The bubble sort will always make one pass which will result in no swaps; after this the test in line 180 will fail and the program will continue with whatever line comes next. I think it is instructive for you to trace exactly what happens to this small routine so that you can see how many times the computer goes through an array in a bubble sort. Fig. 2 is a stack of array elements; the "label" or subscript is shown at the top of the stack (Hmmmmm..... how can you show a subscript at the top? Oh well....) The number at the side represents the value of L1 for the first couple of passes, so you can see the changes line by line.

Boot any DOS disk; enter POKE 104,96 (to put Applesoft programs above both hi-res pages). Type: 10 HOME <CR>
Whoops! what went wrong? (Answer next month - from Reed Harris)

A(1)	A(2)	A(3)	A(4)	A(5)	
7	9	3	1	4	1
7	3	9	1	4	2
7	3	1	9	4	3
ETC.	WHAT COMES NEXT?				

FIG. 2

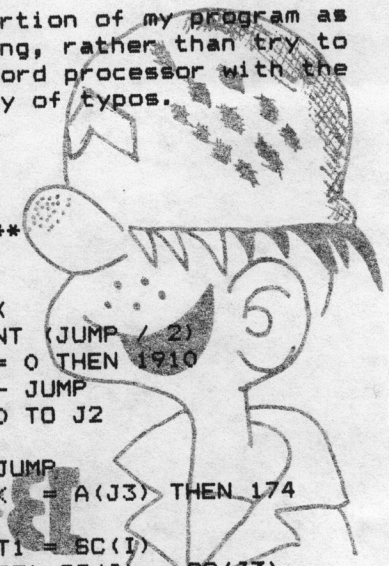
The first pass is filled in; the rest are left for you to work on. The first time through the loop, the 9 and 7 are swapped, the 9 and 3 are swapped, the 9 and 1 are swapped, and then the 4 and 9 are swapped. Because there was a swap (in fact, there were four!) the F = 1 and the test is line 180 is evaluated as TRUE so the entire line is executed, and the loop is started over again. The actual swap is performed by lines 130-150, with T being a temporary variable, since Applesoft doesn't have built-in SWAP function, although there are some versions of BASIC (on minis and mainframes) which do have this ability. Now it's your turn.

Are you back? Did you find that it required three complete passes through the FOR-NEXT loop to complete the sort, with the third pass resulting in no swaps? If you run this program, you will think I have been pulling your leg about the time required, as it runs almost instantly. In order for you to see the results you can add the following lines:

```
200 FOR X=1 TO 5
210 PRINT A(X)
220 NEXT X
```

But change the program by adding a line at the top (5 DIM A(500)), change line 20 to read A(L)=INT(RND(1)*500), delete line 40, and then run it again. First, it will take longer to load the array, but you could have line 40 read PRINT "LOADED!!" so that you can start your timing from the point at which the sort begins. Now you know why there are always efforts made to find a faster sorting algorithm.

The Shell-Metzner sort is not the fastest sort, but it is reasonably easy to study. It is based on comparing two values in different halves of the array, rather than always comparing adjacent elements. Because the Shell sort is rather more complex in its coding, I will give a "dump" of that portion of my program as it is in a listing, rather than try to type it on the word processor with the attendant possibility of typos.



```
1600  REM

** SHELL SORT **


1610 JUMP = MAX
1620 JUMP =  INT (JUMP / 2)
1630  IF JUMP = 0 THEN 1910
1640 J2 = MAX - JUMP
1650  FOR J = 0 TO J2
1660 I = J
1670 J3 = I + JUMP
1680  IF A(I) <= A(J3) THEN 174
     0
1690 T = A(I):T1 = SC(I)
1700 A(I) = A(J3):SC(I) = SC(J3)
1710 A(J3) = T:SC(J3) = T1
1720 I = I - JUMP
1730  IF I > 0 THEN 1670
1740  NEXT J
1750  GOTO 1620
```

FIG. 3

Try a similar trace of this sort routine. You can do this by deleting the bubble sort (lines 100-190) and substituting a GOSUB 1610 with a RETURN at line 1910 (the location of the GOTO in line 1630). Incidentally, notice that the reference to the GOSUB refers to line 1610, the actual starting line, rather than to line 1600 which is the REM line. That way, if you ever decide to pack your program to save memory space, you won't crash the program with a reference to a non-existant line number.

If doing the trace by hand is too old-fashioned for you, try this:

```
165 SPEED = 150
166 FOR J = 1 TO 5:
    PRINT A(J);" ";:
    NEXT J
167 SPEED = 255
```

That way, each change of the array will be listed on the screen. The speed command will slow down the display so that you can read it more easily. 255 is the maximum value, resulting in the normal speed of screen display; 0 is the minimum value in theory, but who wants to sit forever (and I do mean forever!) waiting for the values to print. Line 167 restores the normal speed for the rest of the program.

M.I.T.E.

Version 1.7

Review (c) Copyright by Woody Liswood
April 1983

Published by:

Mycroft Labs, Inc.
P.O. Box 6045
Tallahassee, FL 32301

SYSTEM: CP/M

M.I.T.E is an intelligent terminal program with lot of features. I will leave out the "."s" from now on. **MITE** stands for the "Mycroft Intelligent Terminal Emulator." How about that?

Over the past few years, I have used a variety of terminal programs. I can tell you that currently, **MITE** is my program of choice when communication with THE SOURCE or with any other device which I must address through my DC HAYES modem.

According to the documentation, "Virtually any system that supports ASCII teletypes (or 'dumb' CRT terminals) can be accessed, with full text file transfer capability." I have tried with **APPLES, ESCORTS, OSBORNES**, and two telecommunication networks and have had no problems so I can speculate that their statement is true.

Currently, **MITE** only supports the DC Hayes Micromodem II in the **APPLE** world. I called to ask about other cards and modems and was told that there was little demand for the **APPLE** version of **MITE** so they had not yet worked out the necessary protocols for the variety of cards and modems available. They did say that they would, if demand picked up, add whatever their customers desired.

Seems to me that this is backward marketing, but what do I know? Anyway, this program works great and I hope that they do well so that more devices will become supported.

In addition to the main-line program, **MITE** comes with a set of public domain (read that non supported) utilities which are helpful if you do lots of work with CP/M and telecommunications.

Let me start at the beginning.

INSTALL

The install program works the same as all good programs of this type do. You choose from a variety of terminals and modems. In addition, you can even have a default telephone number. That is a good deal. If you normally only communicate, like I do, with one location. AKA THE SOURCE, then you can install **MITE** with that number and never have to dial again. Other programs use MACRO files to achieve the same effect.

The documentation also contains instructions and hints on installing **MITE** on systems that were not listed in the main menu. However, the instructions expect you to know how to use DDT. If you don't know what DDT is, then the instructions will not make much sense.

MAIN MENU

The main menu appears with a heading telling you what version you have, what the status of the capture buffer is, whether you have bytes captured and a site ID, if you have identified a specific site name. The rest of the menu looks like this:

G - Go Start Communications
H - Hangup Phone
I - Enter Site ID
L - Load Parameters from Disk File
S - Save Parameters on Disk File

P - Parameter Menu
O - Option Menu
U - Text File Upload Menu
D - Text File Download Menu
B - Binary File Transfer Menu
M - Macro String Definition Menu
C - System Command Processor

X - Exit to CP/M

G, when pressed, dials the default telephone number which is contained in the Parameter Menu, or asks you what number you would like to call if there is no number in the parameter menu. Or, if you have set your Role option to ANS, pressing G will wait for the telecommunication link to come through so that communication can start.

I, when pressed, lets you enter a one line site identification. This will automatically be sent to anyone calling **MITE** when **MITE** is in the answer mode.

H - Allows you to hang up your phone. There are some systems, according to the documentation, which will not let you off in any other way.

L, when pressed, allows you to load parameters from a previously saved parameter file. You might like to use this when changing from THE SOURCE to COMPUSERVE. Rather than exiting and rebooting to get a new set of defaults you can just load a parameter file. For example, I saved a parameter file with THE SOURCE phone number. I also saved a parameter file with the COMPUSERVE phone number. If I want to talk with the SOURCE, I boot MITE SOURCE. If I want to talk with COMPUSERVE I boot MITE COMPUSERVE. Then, when the main menu appears I press G and the number is dialed automatically. When I am done with one, I use the L command to change to the new phone number rather than exit and reboot or look up the phone number.

S, when pressed, saves the current parameter status to disk as a parameter file. You would go into the auxiliary menu's and make the necessary parameter changes, then save them as a file from the main menu. Neat and easy.

P gets you to the Parameter menu. This is a sub-menu which contains the following items.

B - Baud Rate	= 300
D - Data Bits	= 7
P - Parity	= EVEN
S - Stop Bits	= 1

R - Role (ANS/ORG)	= ORG
M - Mode (Duplex)	= FULL
N - Phone Number	= 362 6200

X - Exit to Main Menu

The current setting of the parameter is displayed. The program uses two approaches. Some of the menu items toggle between two or three allowable items. Others allow you to enter the data directly. The phone number you see there is the Telenet access number which I use in San Francisco to get to THE SOURCE.

B - for Baud allows you to enter any appropriate BAUD rate. If your equipment, as selected in the install program, does not support that rate you will not be allowed to put that rate in to the program.

D - for data bits toggles between 7 and 8. Most telecommunications networks use 7 (according to the documentation and according to my experience) and most systems which transfer binary (COM) files use 8.

P - for Parity, toggles between NONE, ODD, and EVEN. 7 data bit systems normally use EVEN, and 8 bit systems use NONE.

S - Stop bits toggles between 1 and 2.

R - for ROLE allows you set to be an originator or an answerer of you telephone or direct connect hookup. This is important if you decide to use XTALK or what MITE calls CLINK protocol (more about that later).

M - for MODE goes from FULL to HALF duplex. This is set according to whether the system you are communication with sends back or echos the characters which you send to it. If you are typing and see no characters then the system receiving you is not echoing the characters and you should select half duplex. If you are seeing two characters for each letter you type then you should select full duplex.

N - for number lets you select the phone number you wish to be dialed.

OPTION MENU

The option menu will let you set many special parameters which might be useful.

Here is the menu:

```
E - Escape Trigger Char. = 0AH = ^J
M - Macro Trigger Char . = 1BH = ^[
B - Break Trigger Char.  = 02H = ^B

C - Caps Lock            = OFF
L - Auto LF after CR     = OFF

X - Exit to Main Menu
```

E, for escape trigger, lets you set the character you wish to type so that you can escape from communication with a remote site and return to the main menu to access other parts of the program. This will happen without losing contact with the other system. **MITE** calls this maintaining the transparent link. The documentation goes into a lengthy discussion about using various characters as the escape character. You should notice that the default value is a CONTROL-J not the ESC key. This is because, according to the documentation, many terminals with function keys use a ESC character to precede the function key call.

T, for Macro Trigger, is the key to be type so that a macro can be sent to, or through, the line. It works this way. If a ESC is keyed and the next character is a number from 0 to 9, **MITE** will send the appropriate Macro string. The documentation recommends a Control-L if you are using function keys which already have pre-empted the ESC code.

B, is for the break character. Since I have never used it, let me quote the documentation. "The B option allows you to specify the "Break Trigger" character. Anytime this user-specified character is typed while in the link mode, a communications line BREAK function (SPACE condition for 150 milliseconds) will be perform3d. Note that this is NOT a real character, and is rather out-dated, but is still required by certain older computer systems (mostly IBM)."

C, when on, will only send or receive upper case letters. It works the same as a Caps Lock key on a typewriter.

L, for Automatic Line Feed, is for talking to systems which only send a carriage return character at the end of each line rather than a line feed and a carriage return. I have personally had to use this when talking to some of the bulletin board systems around the countryside.

TEXT FILE UPLOAD MENU

The text file upload menu is, for me, one of the most used features. I am continually send files up to the SOURCE for the PRODUCT REVIEW magazine I publish at PUBLIC 116 DIRECT. Here is how it looks and works.

```
U - Upload Text File
D - Interchar. Delay          = 000
E - Await Char. Echo          = OFF
H - CR/LF Handshaking         = ON
G - Garbage Char. Count       = 000
T - Turnaround Char           = 00H

X - Exit to Main Menu
```

U, for Upload, will ask you the name of the file you wish to send. Upload can be stopped from the keyboard by typing an ESC. You can specify the file name as well as the drive designator. The program assumes XON / XOFF protocol.

D, for Intercharacter delay allows you to slow down the speed of transmission so that a slow system at the other end will have a chance to receive it. This is valuable (in my opinion). I have had to slow down transmissions to send to various systems. Otherwise I would lose almost every third letter or some such crazy thing.

E, for wait for character echo, will cause you to grow a beard when transmitting at 300 baud. It effectively slows the system down to 150 baud because it waits for each character to be echoed back from the other system before sending another character. If you are having transmission problems, then this is a menu item of last resort.

G, for Garbage Character count, will cause this to happen. I have never used this feature so again I will quote from the documentation. "This refers to the number of characters that **MITE** will read and discard at the end of each line before starting to send the next line. Each such "garbage" character has a .5 second timeout. This feature allows the user to upload text to remote systems that send nulls, line numbers, or other extraneous characters at the start of each line before they are ready to receive the new line."

T, for turnaround, will wait for the remote system to send a specific character before a new line will be transmitted. Some timesharing services I have talked to which where associated with schools always send a ? before they will receive data. You could prepare a whole file to be send, specify a ? as the turnaround character then walk away from the transmission.

TEXT FILE DOWNLOAD MENU

Suppose you want to receive a file rather than transmit it to someone. Here is the menu for that function.

```
C - Capture                   = OFF
W - Write Captured Data
R - Reset Capture Buffer
P - Printer Echo              = OFF
T - Type Capture Buffer

F - Flow Control              = OFF
Q - Flow Start Character      = 11H = ^Q
S - Flow Stop Character       = 13H = ^S

X - Exit to Main Menu
```

C turns on the Capture buffer. When you first do this, **MITE** asks you for the file name. If you wish, you can also specify which drive the file is to be placed on. You can also specify any logical CP/M device. If you specify a name which already exists, **MITE** will ask if it is ok to overwrite it. And, if you forget that you have the capture buffer on and exit to CP/M, **MITE** will, then, automatically write the capture buffer to disk for you. That, by the way, is a nice feature which has save some valuable data for me on more than one occasion.

W will cause **MITE** to write the buffer to disk immediately.

R will let you reset the capture buffer to empty.

P turns on or off a printer echo. What this means is that you can save to disk and print at the same time if you also want hard copy. It is a toggle. For it to work, you must have a printer which runs faster than the baud rate you are using to receive data. For example, I have a DIABLO 630 printer which I run at 1200 baud. When using the DC HAYES MicroModem at 300 baud I have had no problems.

T will cause the current contents of the capture buffer to be listed to the screen, 23 lines

at a time.

F, for flow control, is best explained by the documentation itself. "The F option allows you to select whether "flow control" handshaking is to be used while capturing text. This typically refers to XON/XOFF characters. Many systems will pause when you type a control-S and start back up when you type Control-Q. If flow control is enable, **MITE** will attempt to use the mechanism to cause the remote system to pause while it writes to disk the data it has captured since the last pause. This is normally done every 2048 bytes. When the flow stop character is sent out, **MITE** will wait until a full second has elapsed since the last character has arrived before writing to disk . . ."

Q allows you to specify the flow start character.

S allows you to specify the flow stop character.

BINARY FILE TRANSFER MENU

When in CP/M, the difficult thing to do is transfer COM file. There are a multitude of programs and protocols which allow you to do this. **MITE** supports three of those protocols. Here is the menu.

```
P - Protocol                    = XMODEM
S - Send File and Return to Link
R - Receive File and Return to Link

X - Exit to Main Menu
```

P toggles between four protocol used to transmit or receive these types of files. XMODEM is the protocol used on most RCPM systems. This will work with MODEM7, XMODEM, MODEM80, and others which use that type of system.

CLINK is also supported. This, says the documentation, is the original Mycroft Labs binary file protocol. It is compatible with CROSSTALK.

HAYES is the Hayes Terminal Program Verification Protocol. I must say that I have never used it or run across anyone who used it.

IBMPC is the IBM Async Support Package text file protocol.

S allows the files to be sent.

R allows the files to be received.

MACRO STRING DEFINITION MENU

You can store up to 10 Macro strings for use. Each of those strings can be up to 62 characters long. Each of these 10 strings will be stored in any special parameter file you save. And will therefore be loaded through the parameter file load main menu selection. Probably the most normal use for this would be to automate you log on procedure and develop a set of special commands which you use all the time with your favorite system. And, if you define a macro as number 9, it will automatically be sent every time a connection is made. I defined my number 9 macro as my SOURCE log-on procedure. It works.

SYSTEM COMMAND PROCESSOR

The system command processor "understands" many commands which do the same things as regular CP/M commands.

```
DIR  List directory
DRIVE d: Selects a new default drive
ERA  Erases files
REN New = Old  Renames files
RESET makes all drives read/write
Set d:afn $att  gives files new attributes
SPACE D:  shows space on D
TYPE  list file on console
USER n  selects new user number.
```

In using this feature all of the CP/M wildcard conventions are recognized.

EXTRA DATA

There is a good section called "Introduction to Data Communications" in the documentation. You should read it. It contains a discussion of problems of RS232 interfaces, data transmission problems, concepts and what not. The definition of the RS 232 standard and the explainations of what each pin should do was worth the price of the entire program. At least I had in one place a rational readable discussion of problems I had spend some time researching before running across **MITE.**

OTHER SOFTWARE

There are some public domain programs provided on the **MITE** disk. Let me quote: "The following documentation, and the software described therein is included fre of charge (with permission) to save the user from having to download these programs from an RCPM system. The are not supported by Mycroft Labs, but are highly recommended, and appear to work quite well, as documented. They are required for normal use of most RCPM systems, as many text files on these systems are stored in squeezed form, to keep disk storage space and transmission time to a minimum. You may even find them useful in exchanging files with other users. Feel free to share these programs with friends as covered below."

How about that? A publisher that helps rather than hinders. This attitude should be supported.

The programs supplied are:

```
SQ.COM      A file squeezer
USQ.COM     A file unsqueezer
FLS.COM Ambiguous file name expander.
```

The file squeezer compresses files in a compact form so that they can be transmitted faster by a modem and require fewer disks to store or ship programs. The unsqueezer takes the compressed file and returns it to normal. The FLS program expands parameter lists containing wild-card file names into lists with the specific file names required by the Squeezer and the Unsqueezer.

TRSCPM is another program supplied. It is a utility for converting text files from the TRSDOS format to CPM format.

E is also supplied. It is a line-numbered editor program which was created by Mycroft Labs, (according to the documentation) and is similar to line-numbered editors found with BASIC interpreter. I did not test this program.

DOCUMENTATION

The documentation is a full-size three ring binder. It is clear, concise, easy-to-read, makes sense, and generally was a help rather than a hindrance in learning and using the program.

SUMMARY

What can I say. Since working with **MITE** it has become my program of choice when talking to other systems. It works. It is easy to use. The menu's make sense. And, I like the ambiance created by the "free programs." If you need a terminal program, you could not go wrong with this one.

```
########################################
#                                      #
#          AUTHORS WANTED              #
#                                      #
#  Share your experiences, Apple       #
#  programming hints,pascal tips,      #
#  software reviews. Send your         #
#  material to the Cider Press         #
#  Editor at the Clubhouse.....        #
#                                      #
########################################
```

PUNCTUATION
+
STYLE

Review (c) Copyright by Woody Liswood April 1983

Published by:

Oasis Systems
2765 Reynard Way
San Diego, CA 92103

System: CP/M

PUNCTUATION + STYLE are two programs that everyone who writes should use frequently. The check your documents for errors and style.

These programs are produced by the same company who publishes **THE WORD PLUS.** These two programs are companions to the spelling check and correction program.

There are two programs on one disk. The first one I used was called **CLEANUP.** What happens is that **CLEANUP** goes through your documents looking for incorrect, extra or missing punctuation, missing or incorrect space between punctuation, incorrectly spaced ellipsis, incorrect abbreviation, missing capitalization at sentence beginnings, incorrect capitalization, unbalanced quotes, parenthesis, etc. in a paragraph, unpaired underline or **boldface** commands (if you use WordStar), doubled words like the the and numbers which have an incorrect form such as $1,00.00.

See why you need it?

PHRASE is the second program on the disk. It searches your document for words and phrases it does not like and then allows you to do something with or to those words and phrases.

In addition, both of these programs have a number of options which help the user control the output.

Before I get into the programs themselves, I would like to quote from the documentation. Wayne Holder, who wrote the manual has made the manual informative and educational as well as descriptive of the process.

"What "Punctuation + Style" can-t do.

"Cleanup, as a punctuation checker, is not perfect. How to properly punctuate a sentence is sometimes a matter of opinion and not rule. Also, some punctuation decisions require that the checker understand what a sentence means. This is currently impossible for a machine. The purpose of this section is to describe some of these limitations so they won't surprise you.

"Commas are confusing for the novice writer. Commas are often added to show where the reader should pause to properly understand a sentence. CLEANUP cannot tell you where you need a comma to improve understanding. It can point out errors in using a comma, such as a comma preceded by a space or followed by more than one space.

"CLEANUP will tell you when it sees a repeated word such as in "Paris in the the spring." Sometimes the repeated word may be correct such as in "He liked it very, very much."

"Cleanup can often tell you where you need to capitalize the first letter of a word, but it cannot tell you when the first letter should not be capitalized. This is because English allows words to be capitalized in the middle of a sentence if they are used as a proper noun."

INSTALLATION

There is not a standard installation program as you might expect with other CP/M based programs. The documentation states that CLEANUP and PHRASE can read any type of text file. However, the program normally expects to read WordStar files.

There are a set up option files on the disk. If you are using Magic Wand, Spellbinder, Perfect Writer or All Other. These files can be named to over-ride the WordStar defaults in the two programs.

I did test both programs with Spellbinder files as well as WordStar files. The program work the same with both. I did find that CLEANUP did not recognize all spaces at the end of lines in a Spellbinder paragraph so that in many cases CLEANUP told me that I had either too many spaces or not enough spaces at the end of a sentence. Other than that, I found no bugs or problems with either program.

CLEANUP

CLEANUP is run by typing cleanup xxxxxxxx.xxx from the CPM prompt. I actually run it from WordStar using the R command. You do not have to do it that way. You can also rn from the CP/M prompt.

In any case, after you start up the program and after **CLEANUP** locates the first error, you will be shown the error along with an error message such as "This word is repeated. You then have a 3 item menu.

You can mark the spot so that you can return later and fix it. If you do mark the spot, CLEANUP will add the error message to a line below the paragraph which contained the error so that when you return to that spot you can look to see just why it was marked. Just remember to delete that line when you fix your problem.

You can Ignore the problem. This menu item says to **CLEANUP** that you don't agree with its definition of a problem so don't mark this one and go on to the next potential problem.

The final menu item is Quit. If you wish you can quit the marking process at any time. This lets you get away without any marks being made in your file.

CLEANUP marks with a @ sign. You may select a different sign if you wish.

There is also an IGNORE switch which you can use. If you decide while writing your file that you want to set aside a portion of it to be ignored by **CLEANUP** you should surround it with a backslash.

You would do this when you were entering in a set of data which you knew would conflict with the defaults used by CLEANUP and did not want to stop at each "error" or mark each error just so that you could go back and remove those marks.

CLEANUP also has a number of "command line switches." These are extensions to the command line you use to start the program.

A $Q added after the xxxxxxxx.xxx will instruct the program to run "quietly." What that means is that you will not be shown the menu and asked questions. **CLEANUP** will go ahead and mark everything it thinks is a problem and you can then go through your document and make you own decisions.

If you wanted to mark with a * instead of a @ you would add a M* to that command switch.

If you did not want the error messages and marking numbers inserted in your document, you can add a C to that command switch.

Another switch is the F switch. This tells **CLEANUP** to mark the file and place it in a file on your disk. If you had a short file, you could run with the F switch off and just see the errors, but you would not mark the file with the marking character. I can not imagine anyone using that feature. The default is on. Who knows?

The last switch is a D for double spaced documents. Let me quote the documentation: "CLEANUP gets confused when it reads a double-spaced document (one with two carriage returns at the end of each line). Each double-spaced line then looks to cleanup like a separate paragraph. The D switch compensates for this."

The program expect your words to be all lower case, to start with a capital or to be all upper case letters. Anything else will be marked.

PHRASE

PHRASE is the second program on the disk. It reads your document and looks for phrases that match entries in its phrase dictionary. Then it will give you a few hints as to better words or phrases or say that you should just re-write that section.

When I ran **PHRASE**, I discovered that I used the word very too much. And that I like to use the word RATHER a lot of the time. So, I now try not to use the word very. I should know better. Things are or they are not. They are not very.

However, you might be careful here. I did not agree with all of the incorrect usage words or even their suggested changes. One word which I use frequently is Compensation. As a consultant specializing in Human Resources, I do much work with Compensation systems. The phrase dictionary suggest that a better word would be pay. Well, when I want to use Compensation I will, and when I want to use Pay I will. So, I read the dictionary and deleted the word Compensation from it.

As you run **PHRASE** there are four menu items to consider.

First, you can Mark the phrase so that you can return to it at some time in the future and fix it. **PHRASE** marks the entire phrase by enclosing it in [square brackets]. When you correct the phrase, you need to also delete those brackets.

You can also tell the program to Ignore that phrase. It does.

You can have **PHRASE** Print the phrase, sentence and suggestions to your printer. If you do this, you will still have to mark or ignore or suppress the sentence before you can continue.

And, lastly, you can Suppress the sentence. If you suppress the sentence, **Phrase** will remember it and not show it to you again.

SWITCHES

PHRASE also has a set of command switches.

A $P will, with WordStar make each of the suggestions a comment line following the paragraph where the problem occurs.

If you use a $S rather than a $P, **PHRASE** will place the suggestion after the sentence rather than after the paragraph.

There is also the $Q switch. This is the quite switch. When used, **PHRASE** will mark you entire document without stopping and asking your opinion at each problem.

And, as in CLEANUP, **PHRASE** also has the D for Double spaced document switch.

DOCUMENTATION

The documentation is good, readable, educational, informative, and contains the contents of the error and phrase files. In addition, there are instructions for customizing the option files to match up with your particular brand of word processor or text editor.

There is a significant amount of space devoted to discussions about the problems of writing and what **PHRASE** might be able to do for your particular style.

SUMMARY

Another fine program from Oasis. I have used THE WORD PLUS for a long time and was pleased to see the addition of **PUNCTUATION + STYLE** to the set of products. They are useful and necessary tools for anyone

AMERICAN INK PRODUCTS COMPANY
527 Howard Street, San Francisco, California 94105
phone (415) 982-0161

CARTRIDGE RIBBONS

Epson MX 80 ..... @ $5.50
PC 8023 ..... @ $5.00

RECYCLE YOUR USED CARTRIDGES

Refill your own Epson MX 80 or MX 100 ..... @ $2.75
Send your MX 80 or MX 100 to us ..... @ $4.00

RIBBONS FOR ALL BUSINESS MACHINES

Please call us and request prices

These prices are available for members of the San Francisco Apple Core through May 31, 1983

APEX - INFORMATION SYSTEMS INC.
COMPUTER PRODUCTS & SERVICES
Box 11109 • San Francisco, CA 94101
FOR SALES & SERVICE CALL (415) 885-1633

MICROCOMPUTER REPAIR SERVICES

SWIFT ANALYSIS AND REPAIR OF APPLE COMPUTERS
FACTORY TRAINED TECHNICIANS
30 DAY SERVICE WARRANTIES ON ALL REPAIRS
8" CP/M DOWNLOADING AVAILABLE TO MANY DIFFERENT FORMATS

COMPETITIVE RATES

MOST PERIPHERAL CARDS $60 TO REPAIR
DISK DRIVE CALIBRATIONS $40
SERVICE CONTRACTS AVAILABLE (Please inquire)

WIDE RANGE OF EQUIPMENT SERVICED

APPLE II, APPLE IIE, APPLE ///, IBM, ATTACHE
QUME, DIABLO, NEC, MOST APPLE PERIPHERALS
MODEMS, PLOTTERS, MONITORS, POWER SUPPLIES

VARIED SPECIAL SERVICES

CUSTOM PROM BURNING
INSTRUMENT INTERFACING
CORPORATE CONTRACTS AVAILABLE
INSTALLATION AND INTERFACING TO OTHER SYSTEMS

"Apple Breakdown? Call Mike Nadelman!"

A COMMENT ON ERROR TRAPS

By Nick Fotheringham
From The Apple Barrel, July'82

You have finally gotten all of the bugs out of that special program that has kept you in seclusion for the past several weeks. It does exactly what you want it to do, and you are ready to impress someone with it. You beg your boss to take time from his busy schedule for a session with your Apple, and after ten minutes of routine data entry, your program is nearing its flashy finale. The next question appears: "How many sides on an octagon?" As your boss enters "e..i..g...", you stifle, "Not that key, you dummy, the '8'". Too late... The Apple has already responded with a "TYPE MISMATCH" message and shut your program.

One purpose of an "error trap" or "error handling routine" is to help prevent such embarassing situations. Your Apple's BASIC interpreter already has several built-in error traps which were designed to protect the system from your unreasonable requests, such as attempts to divide by zero or to exceed to system's capacity ("STRING TOO LONG", "OVERFLOW", "FORMULA TOO COMPLEX", "OUT OF MEMORY"). Fortunately for many applications, these traps can be avoided by using the ONERR GOTO.....POKE 216,0 commands. ONERR GOTO... disables the system's internal error handling routine and, upon encountering an error, transfers program processing to a statement defined by the GOTO statement, typically a replacement error handling routine of your design. The POKE 216,0 command reinstates the system's error handling routine.

For many beginning programmers, disabling the system's error handling routine, only to replace it with one that you must design and which uses some of your precious RAM memory seems like lunacy. The major reason for doing so is that most of the errors to which the system reacts need not be fatal to your run. The computer views these errors as fatal because the contexts in which they may occur are so diverse that the only general solution that ensures protection to your computer is to terminate your run. However, within your program the context within which an error may occur can often be much more narrowly defined, and nonfatal solutions may be developed. Some of these solutions are described below.

One of the most common applications for error traps is to guard your program against typing errors during data entry from the keyboard. Most such errors can be resolved without aborting your program by designing the program to receive all input as a string variable, say A$. Because A$ will accept input from nearly every key (except RESET) without a TYPE MISMATCH error, it is preferable to A or A% as an input variable. You may then test the input to see if a RETURN has been entered (A$=""), to see if a number has abeen entered (ASC(A$)>47 AND ASC(A$)<58. If the desired numerical input has been entered, you may then convert the input to its numerical equivalent (A=VAL(A$) or A%=INT(VAL(A$))) and then test to see if this value is within the range that you expected as an answer to your question (A%>0 AND A%<5).

One of the great advantages of owning your own computer system on which you run programs interactively is that you can usually train the system to come back to you for help when it has a complaint instead of just dying. When a "fatal" problem is encountered, such as an attempt to divide by zero, an error trap can be used to print an error message of your choosing and then give you an opportunity to change the denominator to a non-zero number and continue the calculation or to abort that program segment (e.g. return to the menu).

Good programs should never "crash". Even when they fail to complete the task for which they were designed, they should reach a controlled ending which provides a detailed description of what went wrong and an opportunity to fix it before ending. Since most of us write programs with the expectation that others will run them, we should get in the habit of using error traps routinely, and we should insist on such programming style in the commercial software we buy.

PASCAL

by
Mike Schnapp

If you are thinking of making the effort to learn a program language, Pascal should be an important consideration. It is a powerful language you should know about and is an ideal first language for programmers. Students who learn Pascal as their first computer language have a relatively easy time learning other languages. Some of the reasons for this consideration are as follows:

(1) Pascal is self-documenting, which may be one of its most important features. Programs written in Pascal are clear and concise. They are made up of English-like sentences that can be arranged so that you can practically read a finished program like a novel or a cook book.

(2) Pascal is a structured computer language that is not ambiguous because it allows the instructions to be grouped into orderly "sections" that are "self-explanatory", in contrast to an unstructured language like BASIC. The programs can consist of an easily digestible series of elements and modules that have clear beginnings and clear endings. There is no question as to which parts of the program do what. GOTO statements are unnecessary. Routines and loops can be organized in understandable packages so that you can work backward or forward through the program without losing the logical sequence or getting lost. Contrast this with programming languages which hinder the creation of self-documenting code (e.g., most versions of BASIC). Comprehending an unstructured program is often like going on a wild goose chase - one part which sends you to another part which sends you...And when you get to the other part, there often isn't any clue as to what is supposed to transpire there! Variable names in BASIC usually are too short to convey any meaning. If there are enough comments, the program may look like a huge undecipherable block with a bunch of line numbers of up to six digits on one side. However, writing a program in Pascal doesn't in any way guarantee that it will be readable, but it is easier to make it so.

(3.) Pascal can be modularized and it allows you to build definitions called PROCEDUREs and FUNCTIONs which can be used in other definitions. These customized modules can be used again and again in later programs.

Today most experts agree that Pascal is on the way to becoming the standard high level language of the computer industry. It may be the most influential contribution to programming made in the past twenty-five years and is not only a programming language but also a statement of programming style. Thus, one major aspect of Pascal is its impact upon the way programs are conceived and composed. In fact the U.S. Department of Defense has adopted a language which is a Pascal descendant, Ada, as the official standard computer language for this country's future computer program development. Pascal contains the very best of several well known languages like BASIC, FORTRAN and COBOL.

If you are interested in joining a group of Pascal users, come to the meetings held in San Francisco on the second and fourth Monday of each month at one of two homes in the Richmond district. Call me at 386-2238 for exact location and further information.

SOFT-STEP
Accent Software
3750 Wright Pl.
Palo Alto, CA 94306

a Review by Joyce Conklin (c) 1983

Soft-Step is a utility for finding bugs in your Applesoft programs. I purposely loaded a program I found in an education magazine which was reputed to have 3 bugs ("Can you find them?"). Well, it sure did - I found five in about 15 lines of program!! But it was a very useful exercise to test out Soft-Step, as I was able to try out all the various modes.
Soft-Step is an assembly language program which resides just below DOS in memory; you load Soft-Step first and then load your BASIC program. The utility is menu-driven; the commands are semi-self explanatory. The copy of the documentation which I received with the review copy was for a previous version - there was a typewritten update enclosed. It is to be hoped that Accent will work on improving the documentation and include a tutorial. This may be a quirk of mine - I like to sit down and read the documentation BEFORE running the program; others I know like to boot the disk and see if they can figure out how it works just from what they see on the screen!
The first capability I tested was the STEP command, which is a fancy form of trace. It will allow you to go through your program one line at a time, but in addition, as each line is executed, you may then PRINT the value of any variable you wish (very useful in finding out why you don't get the "right" answer at the end), or you may PEEK or POKE values directly from or to memory. You may also redefine the value of any variable with a LET command.
This is O.K. the first time, but maybe you don't want to stop at every line, only certain pre-determined ones. In that case you can set specific line numbers, which are stored in a BREAK table, and when you run the program it will go until it reaches the BREAKpoint, at which time it stops and waits.
It is also possible to have the program automatically do certain things at each line; you can have an auto-trace, showing the sequence of executed lines. This is just like TRACE in Applesoft, but much easier to follow, as you can control the speed at which the lines are executed. You may also automatically LIST each line and DO (or execute) each line, which will help you decide what is going on inside your Apple. You can elect to start your program at other than the beginning with the FROM command.
All Soft-Step output is done in INVERSE, which is a bit much at first, but does have the advantage of allowing you to see which is the debugging program and which

is yours. Some cautions are shown in the manual - the display is a scrolling one which may interfere with the way you have decided to have your screen output, especially if you have a display in a reserved area of the screen. You also need to be careful of the dimensioning of arrays; Soft-Step does NOT clear arrays after each run, so if you are testing the program repeatedly, you may get a REDIMENSION ERROR; use CLEAR from Applesoft or C (the clear command) from Soft-Step to avoid this problem. If you are working with a disk access program, there is a warning that setting MAXFILES greater than the default will overwrite Soft-Step.

All in all, if you are writing long Applesoft programs, this is a useful utility, and may indeed allow you to get to bed before 4 a.m. by letting you find your mistake (it's never the computer's mistake!!) and relax -- unless, of course, you are writing against a deadline of 7:30 a.m.!!

Karel the Robot - A Review

A Gentle Programming Teacher

(c) 1982 by Michael A. Fischer

Anyone who has tried to teach computer programming probably has discovered that most languages are too extensive to teach in their entirety. Yet restricting students to a language subset is both over and under inclusive.

In addition, it is often difficult to come up with problems which instruct, build on one another, graphically show their operation, and provide a means for stepping through and debugging a program. In many cases when you try, you also run up against the problems of data structures and disk access techniques.

A solution to these problems is now available. Its name is *Karel the Robot*. [Karel is named after the Czechoslovakian Karel Capek, who first used the word robot in his play R.U.R. (Rossum's Universal Robots). The word robot is apparently derived from a Czech word for forced labor.]

Introducing Karel

The star of this program is, of course, Karel itself, who responds to various commands written in a Pascal-style environment.

Karel lives in a grid which is also populated by walls and beepers. It has a rudimentary sight system able to discern a wall next to it and a hearing system able to hear a beeper if standing next to one. Karel has a soundproof bag which can contain beepers. Finally Karel can tell in which direction it is facing and knows how to turn to its left.

From these simple beginnings Karel can be taught to solve such problems as finding prime numbers.

Programming Book

The book, Karel the Robot, A Gentle Introduction to the Art of Computer Programming, by Richard Pattis, introduces Karel. In well structured chapters it tells you the commands Karel understands and various concepts of programming.

After a general description of Karel's world, the five primitive instructions Karel understands are explained and program structure and debugging are discussed. The technique of defining new instructions from existing primitives (procedures and functions to you Pascal afficianados) is presented along with block structuring and stepwise refinement. Finally conditional execution (If-Then and If-Then-Else) and repetititve instructions (Do and While loops) are explained.

In all cases the book discusses programming philosophy and debugging techniques -- emphasizing the writing of understandable, well-structred programs and full testing and debugging.

At the end of each chapter are a number of problems which can be solved using the techniques and instructions learned so far. The Course disks contain solutions to many of these problems. The book itself contains some suggestions and hints for solutions.

The book, then, is more than a gentle introduction to the art of computer programming. It is an admirable, readable, and concise (100 pages) introduction. And the book is only a part of a comprehensive package.

Simulator

At the heart of the Karel system is the Karel simulator. The simulator comes on two disks which run on the Apple UCSD Pascal System. The program itself is copy protected and appears on both disks.

There are three parts to the simulator. The first part is a "compiler" which takes programs written in Karel source code (using the Pascal text editor) and compiles it, priting to the screen each instruction as it is compiled and the level of nesting it represents.

When a problem is encountered the compiler stops, shows where the error is, and suggests what you might have meant. You then use the Pascal editor to correct the mistake. All compiler output is displayed on

the monitor screen. It would be helpful if the display could optionally be sent to a printer.

World Builder

The second part of the simulator is the world builder. It is used to create the initial physical situation in which Karel is placed. By moving Karel around with simple commands you can build walls, place any number of beepers on various corners and in Karel's beeper bag, and leave Karel on a particular corner facing in a particular direction for its initial situation. A command summary can be called from the world builder by pressing "?".

The third part of the simulator -- the run time system -- is where the genius of the program really shows. There are three execution modes: (1) automatic which executes to completion without user intervention; (2) monitor which supports fairly sophisticated debugging facilities; and (3) explain which shows in great detail each step being executed, each test made (If-Then-Else, While, and Iterate) and the result of that test.

The run time system displays the portion of Karel's grid containing Karel and the current status. The status line is not updated when the simulator is run at the fast speed option. At medium speed (slower than fast by a factor of five) the status lines show the location, direction Karel is facing, number of beepers in the bag and on the current corner, number of instructions executed, and current defined instruction being executed. [A defined instruction is one which is part of the main program.] At slow speed (slower by another facotr of five) the current primitive instruction being executed as part of the current defined instruction is also shown on the status lines.

Display Modes

The run time system also permits two display modes -- regular which updates the screen after each move Karel makes, and "sticky" which leaves an image of Karel on each corner it has been on, facing the the last direction it faced on that corner.

After Karel has finised execution you have five options: (1) exit to the Pascal system; (2) save the current world as a world file; (3) view a part of the world not being currently displayed; (4) reexecute the program on the same or a different world; or (5) backtrace through the program.

Running Karel in the monitor mode provides access to a sophisticated program debugger. You can change execution speeds and move forward or backward one or more steps at a time. The definition of any defined instruction can be recalled to the screen. Finally breakpoints can be set so exeuction can continue to a specified instructions or primitive instruction level.

Done with Care

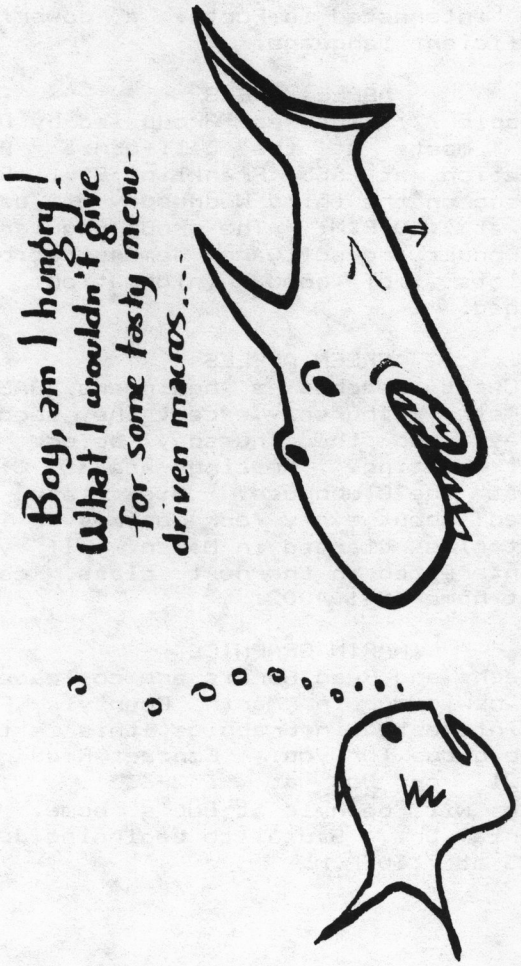
The Karel run time system has several attributes which show the care with which it was developed. In the more complex modes typing "?" will call up a screen of instructions. And whenever a user answer is called for, the default is both shown and carefully thought out to be a common or usual response. In fact, the only real complaint I have about the program -- and one which is admittedly reaching a bit -- is the lack of any facility for generating hardcopy output.

In addition to the two simulator disks there are two course disks which contain many examples and problem solutions already entered as both text files and world situations.

The system is an excellent teaching tool either for self-teaching, home teaching, or especially for the ever-increasing computer courses in schools at all levels.

The book itself is available for $7, the simulator (in either a 40 or 80 column version) for $85, and the course disks for $150. The products are available from:

Cybertronics International, Inc.
999 Mt. Kemble Avenue
Morristown, N.J. 07960
(201) 766-7681



INTEREST GROUPS

CP/M

For those members interested in learning about Apple CP/M there is an interest group in Marin led by Jim Ayers. The group meets at 411 Ash St., Mill Valley, the second and fourth Tuesday of the month at 7:30 P.M. Jim reports that the meetings are generally question and answer sessions and that there is no formal structure to the meeting.

EAST BAY

Molly Mitchell and Leonard Leff are co-leaders of the East Bay Group which meets at 766-4th St., Oakland, at 7:30 P.M. on the first Tuesday of the month. This unstructured group is generally for those members interested in exchanging general information about their apples.

FORTH

Dudley Ackerman leads an informal meeting regarding the Forth language on the first and third Tuesday of the month at 7:30 P.M. at the clubhouse. Dudley welcomes anyone interested in Forth, a powerful and efficient language.

APPLE ///ERS

The Apple /// Interest Group led by Don Norris meets at the California Bar Association at 555 Franklin St., San Francisco on the third Wednesday of each month at 7:30 P.M. The group regularly has product and software demonstrations and lots of good information is exchanged.

GREEN APPLES

Gary Owens teaches a beginners BASIC class every Thursday except the second Thursday and the Thursday before a general meeting. Meetings are at 8:00 P.M. at the Clubhouse. His class is recycled about every four monthes. The current class started in March. If you are interested in the next class, call Gary at home (541-4002).

MARIN GRAPHICS

Bob Sachs and Reed Harris are co-leaders of a new group in Marin County. For those interested in Graphics, this is the is the group for you. Contact Reed at 924-7221 or Bob at 332-3331. The meetings will be held at Bob's home, 41 Cresienta Dr., Sausalito beginning June 8, 1983 at 7:30 P.M.

NU'ERS

The Nu'ers (New Users) is a interest group dedicated to helping new users. Monthly a new topic will be discussed that will help new users become more familiar with their Apples. In May the topic will be "What The Dealer Won't or Can't Tell You and the Manuals Garble." In June we will probably explore the printer! The meetings are held on the second Tuesday of the month at 7:30 P.M. at Guinness Productions, 55 Main St., Tiburon. Contact Sami Capron for further information (388-8631).

TELECOMMUNICATIONS

For those interested in using a modem a new group has been formed that will meet at the Clubhouse on the fourth Wednesday of the month at 7:30 P.M. The first meeting will be held on May 25, 1983 and the group will look at the Source and discuss the future trend of the group. For further information contact the group leader Frank Fennelly, (408) 984-7321.

dBASE II

A new group has been formed for those interested in learning how to use dBase II. Shirley Anderson, the group leader, has been negotiating with Firemans Fund in Marin for the use of their computers! It is reasonably certain that the group will meet at Firemans Fund, 1 Lucas Valley Rd., San Rafael on the third Thursday of the month beginning in June at 7:00 P.M. Contact Shirley for more information (892-0522).

EDUCATION ANNOUNCEMENTS

Nibble Counting:John Uhley will be offering a class in nibble-counting. It will be given at the Marin Computer Center, 50 El Camino Dr. in Corta Madera on five Saturday afternoons, May 14, 21, 28, June 4 and 11, from 2-4 P.M. There will be a charge of $85.00 for the course. For further information contact John at 461-7626.

EPSON

The Epson printer interest group meets on the fourth Tuesday of each month at the clubhouse, 1515 Sloat Blvd. Meetings begin at 7:30 P.M. and usually last until 9:00 or 9:30 P.M. Contact Michael Fischer for further information. (933-4435)

MAY 1983

MONDAY	TUESDAY	WEDNESDAY	THURSDAY
	3 East Bay 7:30 Forth 7:30		5 Green Apples 8:00
9 Pascal 7:30	10 CP/M 7:30 VisiCalc 7:30 Nu'ers 7:30	11 Board of Direc- tors 7:00	
	17 Forth 7:30	18 Apple /// 7:30	
23 Pascal 7:30	24 CP/M 7:30 Epson 7:30	25 Telecommunica- tions 7:30	26 Green Apples 8:00

JUNE 1983

			2 Green Apples 8:00
	7 East Bay 7:30 Forth 7:30	8 Marin Graphics 7:30 Bd. of Directors 7:00	
13 Pascal 7:30	14 CP/M 7:30 VisiCalc 7:30 Nu'ers 7:30		
	21 Forth 7:30	22 Telecommunica- tions 7:30	23 dBASE-II 7:00 Green Apples 8:00
27 Pascal 7:30	28 CP/M 7:30 Epson 7:30		30 Green Apples 8:00

INTEREST GROUP LEADERS

Apple ///	Don Norris	673-7635 (day)
CP/M	Jim Ayers	383-0473(day)
		383-5759(eves)
dBASE II	Shirley Anderson	892-0522(home)
Epson	Michael Fischer	557-1250 (day)
		933-4435(eve)
Forth	Dudley Ackerman	626-6295
Green Apples (BASIC)	Gary Owens	521-4002
Marin Graphics	Reed Harris	924-7221
	Bob Sachs	332-3331
Nu'ers	Sami Capron	388-8631
Pascal	Mike Schnapp	386-2238
Product Review	"Woody" Liswood	585-6375
Telecommunications	Frank Fennelly	(408)984-7321
VisiCalc	Don Scellato	672-2051(eve)

APPLE CORE OFFERINGS

NOTES

SFAC 1983 DISK NUMBER 1 contains only three programs due to the nature of one of the programs - a hi-res graphics game. This game is called "The Incredible Shrinking Man" which was contributed by the Chiang Brothers. Also included on the disk is a High Score Tabulator and hi-res graphics display program called Gallery.

LIBRARY #1 contains over 340 programs. It is packaged in diskette holders and is bound in a SF Apple Core binder with over 30 pages of documentation. This documentation alone is available for three dollars.

LIBRARY #2 continues the tradition started with Library #1. It contains over 250 programs from the 1980 DOMs.

1981 YEARBOOK contains 12 disks containing all DOM programs from 1981. There is a heavy emphasis on games and business applications.

1983 PROGRAMMING UTILITIES contains a collection of utilities not collected on a single disk before. These utilities come from a variety ofsources such as IAC disks, 1981 DOMs, and 1982 DOMs.

1983 GAME SET consists of two disks containing games not collected in a single set before. It contains a variety of hi-res, lo-res, and text games.

1983 EDUCATION APPLICATIONS is a set of two disks containing interactive learning programs, a grade maintenance program, and GRAMMAR QUEST an interactive learning game.

1983 HOME APPLICATIONS consists of two disks of programs useful around the home. One disk was from the 1982 Home Pak. The second disk is a collection of programs gathered from IAC disks and other recent contributions.

Instructions: The easiest way to get these products is to visit the 'sales table' during any General Meeting. Otherwise, circle the items you wish to purchase, place this sheet into an envelope with check or money order (DO NOT send cash!) to:

San Francisco Apple Core, Inc.
1515 Sloat Blvd., Suite #2
San Francisco, CA 94132
Attn:Marketing Dept.

PRODUCT	MTG	MAIL
Library #1	98.00	101.50
Library #2	98.00	101.50
1981 Yearbook	98.00	101.50
DOMs		
Mar 82	5.00	7.00
Apr 82	5.00	7.00
May 82	5.00	7.00
COLLECTIONS		
Basic Tutor Pak(3)	20.00	22.00
Forth Pak (3)	20.00	22.00
Game Set (2)	15.00	17.00
Home Applications(2)	15.00	17.00
Education Applic.(2)	15.00	17.00
1983 Prog Util.(1)	10.00	12.00
SFAC Visicalc Temp(1)	10.00	12.00
1983 Visicalc Util(1)	18.00	22.00

All foreign orders: Add 25% to above prices.

SFAC MAIL ORDER BLANK

NAME.....

ADDRESS.....

CITY.....STATE.....

ZIP.....

SPIN-TRONICS SAVES YOU MONEY

DISK DRIVES	
Rana Elite 1	259.00
Quentin	279.00
Apple	340.00
Diskettes	
Elephant ss/sd	22.00
Elephant ss/dd	24.00
Elephant ds/dd	29.00
Verbatim ss/sd	26.00
Wabash ss/sd	19.00
Flip 'n File	23.00
MODEMS	
Micromodem II	269.00
Smartmodem	210.00
Apple-Cat II	278.00
J-Cat	123.00
SSM ModemCard	258.00
HARDWARE	
Snapshot	99.00
Wild Card II	105.00
Wizard-80	160.00
Parallel Intf.	70.00

PRINTERS	
Dot Matrix	235.00
Gemini 10	335.00
Prowriter I	425.00
Prowriter II	650.00
Microline 92	525.00
Daisywriter	1075.00
Paper	20.00
MONITORS	
BMC 12AU	95.00
Video 300 Ambr.	165.00
USI Pi-3	160.00
Taxan 12" Ambr.	135.00
Taxan 12" Grn.	127.00
Taxan RGB	308.00
Comrex Color	340.00
Amdek Color 1	318.00

MAIL & PHONE ORDERS ACCEPTED. PLEASE ADD $4.00 FOR SHIPPING.

Send $1.00 (refundable) for complete catalog.

SOFTWARE	
Visicalc Enhanced	190.00
PFS-File for IIe	95.00
Screenwriter II	89.00
Word Handler	140.00
MultiPlan	215.00
Tax Preparer	152.00
Tax Manager	136.00
Tax Advantage	45.00
Miner 2049'er	30.00
Zaxxon	30.00
Tubeway	26.00
Mask of the Sun	30.00
Frogger	26.00
Dark Crystal	30.00
Copy II plus	27.00

SPIN-TRONICS
2490 Channing Way
Suite 409
Berkeley, CA 94704
(415) 843-2743
at Telegraph

INTERNATIONAL APPLE CORE™

APPLE ORCHARD SUBSCRIPTIONS

For IAC Club Members Only

908 GEORGE STREET, SANTA CLARA, CALIFORNIA 95050

☐ 9 issues for only $19.50
($29.25 at newsstands)
$17.00 for members of IAC clubs only

☐ Foreign: 9 issues
$36.00 Surface
$29.00 for members of IAC clubs only

☐ Payment enclosed
(US funds only)

Start my subscription with issue # _____

Name _____

Address _____

City _____ State _____ Zip _____

☐ Name of Club _____ Signature of club officer _____
Valid only with signature

InterBank # ☐☐☐☐ 4 digit number above your name on MasterCard™ ☐ VISA

Charge Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

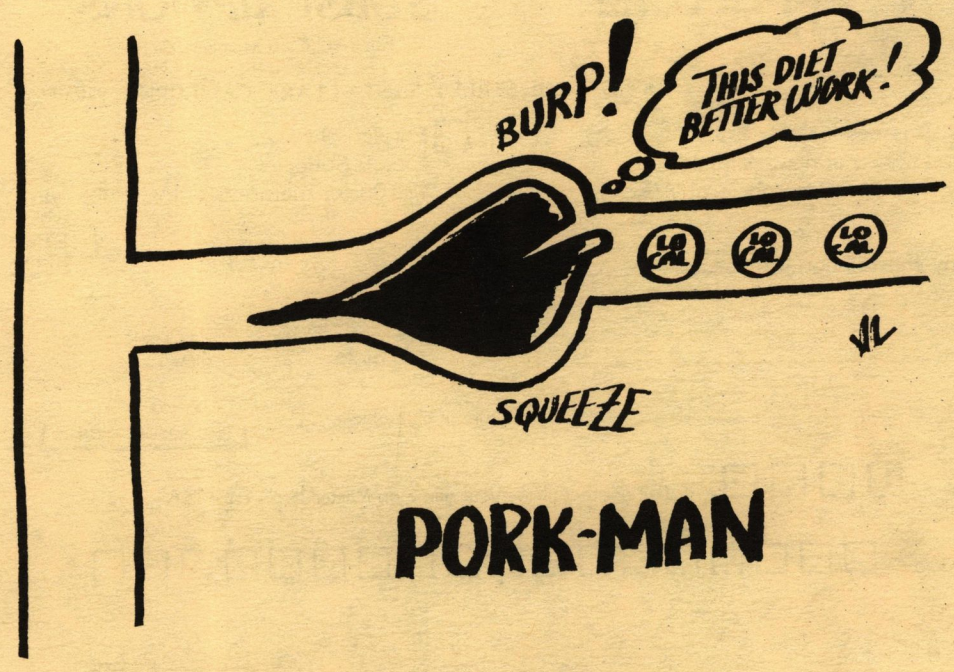
_____ _____
Credit Card Signature Card Expires

SF APPLE CORE
1515 Sloat Blvd., Suite 2
San Francisco, CA. 94132

BULK RATE
U.S. POSTAGE
PAID
Permit No. 290
Concord, CA 94520

TIME VALUE

TO:



PORK-MAN